Web-based entity search and analysis of Social Network

Modelling the data:

The dataset and relationships among the entities was easy to interpret. After understanding the data, I determined that this was an ideal use case for a graph database. Storing this data on a graph database would allow an efficient way of querying the components based on their attributes as well as relationships with other components.

Having no prior experience working with a graph database, my biggest challenge was learning this concept and implementing this data model within the given time frame. I started by reading Neo4j's 'Getting started guide' to learn about all components and high-level architecture of their graph database.

I then went on to learn 'cypher', by completing their guided tutorial on Neo4j Sandbox. Once I was ready, I went on to create a cloud instance of a graph database on Neo4j Aura.

I installed the Neo4j driver for python and followed its documentation to connect to the database and execute queries.

Developing the web app:

On seeing the data, I was keen on developing the entity search feature, which could be extended to perform entity resolution and record linkage in larger datasets where duplicate entities may exist with minor inconsistencies among them. In my consultation with Mr. Nikhil Almeida, I discussed my plan to deliver this feature. I saw it best to implement it is as a web interface where user can input all search parameters to find matching entities.

Additional things I would have liked to include in the application:

1. Allow user to choose minimum similarity index to pass as fuzzy match
   The apoc library provides a series of functions like apoc text levenshtein, apoc text hamming, etc to calculate string similarity. This can be compared against the threshold set by user and return results accordingly
2. Visualization
   An interactive visualization section would be a great way to provide a quick look of how the structure of the network looks, how the nodes are grouped and related.
3. Analysis section
   - While identifying cliques in the network, we may apply constraints of relationship type or similar type of entity. While in the given dataset, such analysis would not have produced significant insights but provides a great way to identify tightly related components in a large dataset.

- Additionally, I would have liked to implement the local cluster coefficient algorithm to determine subsets of entities which are close to becoming a clique. In such a network, it can provide a way for us to estimate hidden/possible relationships or predict new relationships.

General thoughts and key takeaways:

Overall, this challenge was a great learning experience. Through this project, I came across and worked on technologies and concepts that I have never worked with before.

It has made me confident that with more time on hand, I can make this application better, to suit larger datasets and enhance the features to -

a.) perform deeper analysis and draw valuable insights about the components of the network
b.) giving user the option to tune certain parameters of analysis, to meet their needs better
c.) visualize the results in a more interactive manner

With this experience, I can see myself working to learn and experiment more with graph databases and analysis on graph databases across different use cases.